

# UNO 24



Created by

Chomchaba Chunakiat 6431308021

2110215 Programming Methodology

Semester 2 Year2021

Chulalongkorn University

## Introduction

เกม UNO 24 มีเค้าโครงมาจากเกมการ์ด UNO ที่มีอยู่แพร่หลายทั่วไป โดยมีการดัดแปลงบางส่วนเพื่อเพิ่มความสุขให้กับเกมมากขึ้น ในเกมประกอบด้วยการดัดชนิดและสีต่างๆ ตอนเริ่มต้นจะแจกการ์ดให้กับผู้เล่นแต่ละคน โดยจะมีการ์ดหนึ่งใบวางอยู่บนโต๊ะตรงกลาง ผู้เล่นแต่ละคนจะต้องผลัดกันลงการ์ดที่มีสีหรือชนิดเดียวกับการ์ดบนโต๊ะ การ์ดที่ลงไปจะไปเป็นการ์ดบนโต๊ะใบใหม่สำหรับผู้เล่นคนต่อไป ผู้เล่นที่ลงการ์ดจนหมดก่อนจะเป็นผู้ชนะในรอบนั้น ผู้ชนะในแต่ละรอบจะได้แต้มเท่ากับจำนวนการ์ดที่เหลือในมือของผู้เล่นคนอื่น ผู้เล่นที่สะสมแต้มได้ครบ 24 แต้มก่อนจะเป็นผู้ชนะ เกม UNO 24 มีผู้เล่น (Player) ทั้งหมด 4 คน เป็นคนเล่น 1 คน อีก 3 คนเป็นบอทที่เขียนโปรแกรมให้เล่นเองโดยอัตโนมัติ

## Game Controls

### 1. Home Screen



Figure 1: Home Screen

Home Page เป็นหน้าแรกของเกม Home Page ประกอบด้วย 3 ปุ่มหลัก ดังนี้

- ปุ่ม START : กดปุ่มนี้เพื่อเริ่มเล่นเกม กดแล้วจะมีหน้าต่างใหม่แสดงขึ้น ให้ตั้งชื่อ Player หลังจากนั้นกดปุ่ม OK เพื่อนำไปสู่หน้าเกม กดปุ่ม Cancel เพื่อยกเลิกการตั้งชื่อและปิดหน้าต่าง
- ปุ่ม HOW TO PLAY : จะแสดงหน้าต่างย่อยที่มี link ของคู่มือเกม
- ปุ่ม QUIT : กดปุ่มนี้เพื่อออกจากเกม

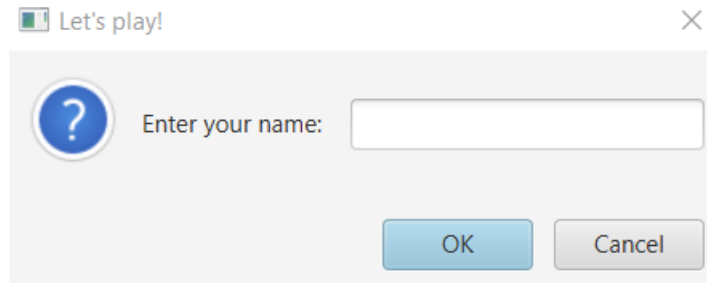


Figure 2: Name Setting Window

## 2. Game Screen

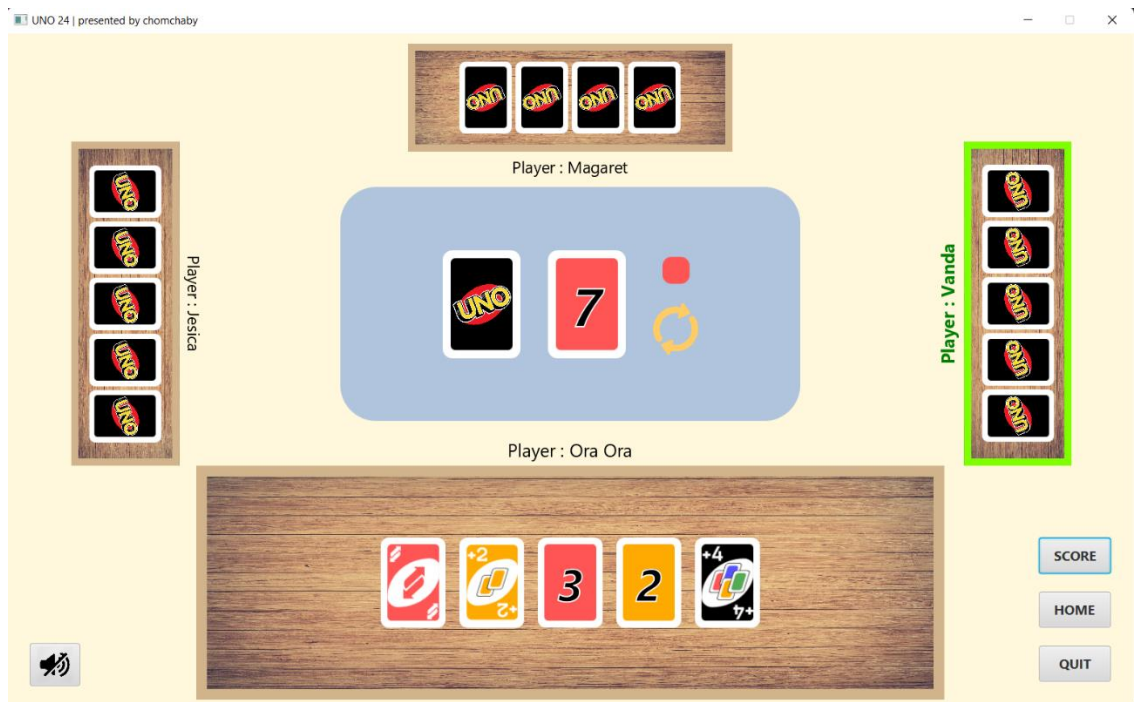


Figure 3: Game Screen

ตอนเริ่มต้น แต่ละ Player จะได้รับการ์ดทั้งหมด 5 ใบ โดยจะสุ่มผู้เล่นที่เริ่มเล่นเป็นคนแรก แต่ละ Player จะมีกระดานการ์ดของตัวเองและมีชื่อแสดงไว้ข้างบนดังรูป กระดานข้างล่างที่มีขนาดใหญ่สุดเป็นของคนเล่น ที่เหลืออีก 3 กระดานเป็นของบอท โดยมีชื่อ Jessica, Magaret และ Vanda เมื่อถึงตาของผู้เล่นคนไหน จะมีกรอบสี่เหลี่ยมรอบกระดาน เช่นตัวอย่างในรูปที่ 3 เป็นตาของ Vanda



Figure 4: กระดานคนเล่น (User Pane)



Figure 5: กระดานบอท (Bot Pane)

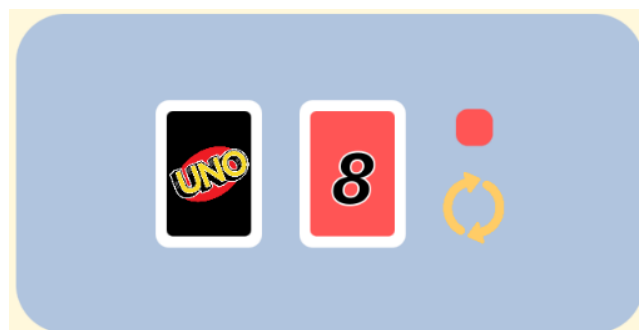


Figure 6: โต๊ะตรงกลาง (Center Pane)

โต๊ะสี่ฟาดตรงกลาง (Center Pane) แสดงการ์ดที่เปิดอยู่บนโต๊ะ กองการ์ดสำหรับจั่ว เครื่องหมาย แสดงทิศการวนผู้เล่น และกล่องสี่แสดงสถานะสีของการ์ดที่อยู่บนโต๊ะ

### 3. How to play

ผู้เล่นจะต้องลงการ์ด 1 ใบ ให้สอดคล้องกับสีหรือตัวเลขหรือสัญลักษณ์พิเศษบนการ์ดที่วางไว้บน โต๊ะ หรือลงตามข้อกำหนดของการ์ดก่อนหน้านี้ โดยการ์ดแต่ละประเภทจะมีเงื่อนไขการลงที่แตกต่างกัน ตามประเภทและความสามารถของการ์ดใบนั้น

ในกรณีที่ผู้เล่นไม่มีการ์ดที่สามารถลงได้จะต้องจั่วการ์ดจากกองกลางขึ้นมา 1 ใบ ถ้าการ์ดที่จั่วได้ สามารถลงได้ ผู้เล่นจะต้องลงการ์ดใบนั้นจึงจะจบรอบการเล่น ถ้าการ์ดที่จั่วได้ไม่สามารถลงได้จะจบรอบ การเล่นในรอบนั้นและผู้เล่นคนถัดไปจะได้ลงต่อ

ในกรณีที่ผู้เล่นมีการ์ดที่สามารถลงได้ผู้เล่นจะต้องลงการ์ด 1 ใบ ถ้าผู้เล่นไม่ต้องการลงการ์ดใบ นั้นสามารถจั่วการ์ดเพิ่มจากกองกลางได้ 1 ใบ หลังจากนั้นผู้เล่นจะต้องลงการ์ด 1 ใบ เพื่อ จบรอบการเล่นในรอบนั้น

โดยกรอบของกระดานการ์ดผู้เล่นปกติจะเป็นสีน้ำตาลอ่อนและจะเปลี่ยนเป็นสีเขียวเมื่อถึงตา ของผู้เล่นที่ต้องเล่น ถ้ากรอบขึ้นเป็นสีแดงแสดงว่าผู้เล่นคนนั้นถูกการ์ดหยุดจากผู้เล่นคนก่อนหน้าและไม่สามารถเล่นในรอบนั้นได้

### ประเภทของการ์ดในเกม

#### 1. การ์ดตัวเลข

การ์ดตัวเลขมีทั้งหมด 80 ใบ ประกอบด้วยการ์ดตัวเลขตั้งแต่เลข 0 – 9 มี 4 สี คือ แดง เขียว น้ำเงินและเหลือง โดยมีสีละ 2 ใบ โดยการ์ดตัวเลขจะสามารถลงได้เมื่อการ์ดนั้นมีสีเดียวกันหรือมีเลข เดียวกันกับการ์ดที่ลงไว้ก่อนหน้านี้



Figure 7: การ์ดตัวเลข (Normal Card)

## 2. การ์ดวนกลับ

การ์ดวนกลับเป็นการ์ดที่มีความสามารถพิเศษ เมื่อผู้เล่นใช้การไบนีจะทำให้ทิศทางการวนเปลี่ยนกลับเป็นอีกทาง โดยการ์ดวนกลับมีทั้งหมด 8 ใบ แบ่งเป็น 4 สี คือ แดง เขียว น้ำเงินและเหลือง โดยมีสีละ 2 ใบ โดยการ์ดวนกลับจะสามารถลงได้เมื่อการ์ดก่อนหน้านี้มีสีเดียวกันหรือเป็นการ์ดวนกลับเหมือนกัน



Figure 8: การ์ดวนกลับ (Reverse Card)

## 3. การ์ดหยุด

การ์ดหยุดเป็นการ์ดที่มีความสามารถพิเศษ เมื่อผู้เล่นใช้การไบนีจะทำให้ผู้เล่นที่จะต้องเล่นเป็นคนถัดไปไม่สามารถเล่นในรอบนั้นได้ ซึ่งจะข้ามให้ผู้เล่นคนถัดไปเป็นผู้เล่นต่อ โดยการ์ดหยุดมีทั้งหมด 8 ใบ แบ่งเป็น 4 สี คือ แดง เขียว น้ำเงินและเหลือง โดยมีสีละ 2 ใบ โดยการ์ดหยุดจะสามารถลงได้เมื่อการ์ดก่อนหน้านี้มีสีเดียวกันหรือเป็นการ์ดหยุดเหมือนกัน



Figure 9: การ์ดหยุด (Skip Card)

#### 4. การ์ดบวกสอง

การ์ดบวกสองเป็นการ์ดที่มีความสามารถพิเศษเมื่อผู้เล่นใช้การ์ดใบนี้จะทำให้ผู้เล่นที่จะต้องเล่นเป็นคนถัดไปได้รับการ์ดจากกองกลางเพิ่มอีก 2 ใบโดยอัตโนมัติ โดยการ์ดบวกสองมีทั้งหมด 8 ใบ แบ่งเป็น 4 สี คือ แดง เขียว น้ำเงินและเหลือง โดยมีสีละ 2 ใบ การ์ดบวกสองจะสามารถลงได้เมื่อการ์ดก่อนหน้านี้ไม่มีสีเดียวกันหรือเป็นการ์ดบวกสองเหมือนกัน



Figure 10: การ์ดบวกสอง (Draw Card)

#### 5. การ์ดชาเลนจ์

การ์ดชาเลนจ์เป็นการ์ดที่มีความสามารถพิเศษ เมื่อผู้เล่นใช้การ์ดใบนี้จะทำให้ผู้เล่นที่ลงการ์ดต้องทายว่าผู้เล่นคนถัดไปไม่มีการ์ดสีอะไร ถ้าผู้เล่นที่ลงการ์ดทายถูก จะทำให้ผู้เล่นคนถัดไปได้รับการ์ดเพิ่มจากกองกลาง 4 ใบ โดยอัตโนมัติ แต่ถ้าผู้ลงการ์ดทายผิดจะทำให้ผู้ลงการ์ดได้รับการ์ดเพิ่มจากกองกลางเพิ่มอีก 2 ใบ โดยอัตโนมัติ และสีที่ผู้ลงการ์ดเลือกทายจะเป็นสีที่กำหนดสีของการ์ดที่สามารถลงได้ในครั้งต่อไป ซึ่งการ์ดชาเลนจ์จะมีทั้งหมด 6 ใบ โดยการ์ดชาเลนจ์จะสามารถลงได้โดยที่ไม่จำเป็นต้องมีสีหรือสัญลักษณ์เหมือนการ์ดก่อนหน้านี้ เนื่องจากเป็นการ์ดสีดำ



Figure 11: การ์ดชาเลนจ์ (Challenge Card)

#### 6. การ์ดเลือกสี

การ์ดเลือกเป็นการ์ดที่มีความสามารถพิเศษ เมื่อผู้เล่นใช้การ์ดใบนี้จะทำให้ผู้เล่นที่ลงการ์ดสามารถเลือกได้ว่าจะให้ผู้เล่นคนถัดไปลงการ์ดที่มีสีอะไร ซึ่งการ์ดเลือกสีจะมีทั้งหมด 4 ใบ โดยการ์ดเลือกสีจะสามารถลงได้โดยที่ไม่จำเป็นต้องมีสีหรือสัญลักษณ์เหมือนการ์ดก่อนหน้านี้เนื่องจากเป็นการ์ดสีดำ







## Implementation Detail

### 1. Package entity.card

#### 1.1 Interface Actionable

Interface ระบุ method ของการ์ดที่มีความสามารถพิเศษ

##### 1.1.1 Method

Name	Description
+ void takeAction()	ถูกเรียกเมื่อการ์ดถูกทิ้งลงบนโต๊ะ

#### 1.2 Abstract Class UnitCard

##### 1.2.1 Field

Name	Description
- int number	เลขประจำการ์ด บ่งบอกชนิดของการ์ดได้
- Color color	สีของการ์ด

##### 1.2.2 Constructor

Name	Description
# UnitCard(int number, Color color)	ตั้งค่า field number ด้วย number ตั้งค่า field color ด้วย color

##### 1.2.3 Method

Name	Description
+ <i>boolean isPlaceable()</i>	บอกว่าลงการ์ดบนโต๊ะได้หรือไม่
+ <i>String toString()</i>	คืนค่าข้อความบอกลักษณะการ์ด
+ <i>String myColorToString(Color color)</i>	คืนค่าข้อความบอกชื่อสี
+ GETTER for each field	

### 1.3 Class NormalCard extends UnitCard

#### 1.3.1 Constructor

Name	Description
+ NormalCard(int number, Color color)	เรียก constructor จาก superclass

#### 1.3.2 Method

Name	Description
+ boolean isPlaceable()	- ลงได้เมื่อสีหรือตัวเลขตรงกับการ์ดบนโต๊ะ - ลงได้เมื่อสถานะสีของการ์ดบนโต๊ะเป็นสีดำ
+ String toString()	คืนค่า Integer.toString(getNumber()) + " (" + UnitCard.myColorToString(getColor()) + ")"

### 1.4 Abstract Class EffectCard extends UnitCard implements Actionable

#### 1.4.1 Constructor

Name	Description
# EffectCard(int number, Color color)	เรียก constructor จาก superclass

### 1.5 Class SkipCard extends EffectCard

#### 1.5.1 Constructor

Name	Description
+ SkipCard(Color color)	เรียก constructor จาก superclass โดยให้ number เท่ากับ 10

#### 1.5.2 Method

Name	Description
+ boolean isPlaceable()	- ลงได้เมื่อสีหรือตัวเลขตรงกับการ์ดบนโต๊ะ - ลงได้เมื่อสถานะสีของการ์ดบนโต๊ะเป็นสีดำ

+ void takeAction()	- ถ้าผู้เล่นคนปัจจุบันเป็นบอท หยุดการเดินเกม 1 วินาที ตั้งค่า isPlayable ของผู้เล่นคนต่อไปเป็น false
+ String toString()	คืนค่า "Skip Card" + " (" + UnitCard.myColorToString(getColor()) + ")"

## 1.6 Class ReverseCard extends EffectCard

### 1.6.1 Constructor

Name	Description
+ ReverseCard(Color color)	เรียก constructor จาก superclass โดยให้ number เท่ากับ 11

### 1.6.2 Method

Name	Description
+ boolean isPlaceable()	- ลงได้เมื่อสีหรือตัวเลขตรงกับการ์ดบนโต๊ะ - ลงได้เมื่อสถานะสีของการ์ดบนโต๊ะเป็นสีดำ
+ void takeAction()	- ถ้าผู้เล่นคนปัจจุบันเป็นบอท หยุดการเดินเกม 1 วินาที สลับทิศการวนผู้เล่น
+ String toString()	คืนค่า "Reverse Card" + " (" + UnitCard.myColorToString(getColor()) + ")"

## 1.7 Class DrawCard extends EffectCard

### 1.7.1 Constructor

Name	Description
+ DrawCard(Color color)	เรียก constructor จาก superclass โดยให้ number เท่ากับ 12

### 1.7.2 Method

Name	Description
+ boolean isPlaceable()	- ลงได้เมื่อสีหรือตัวเลขตรงกับการ์ดบนโต๊ะ - ลงได้เมื่อสถานะสีของการ์ดบนโต๊ะเป็นสีดำ

+ void takeAction()	- ถ้าผู้เล่นคนต่อไปเป็นคน หยุดการเดินเกม 2 วินาที ผู้เล่นคนต่อไปจั่วการ์ด 2 ใบ
+ String toString()	คืนค่า "Draw Card" + " (" + UnitCard.myColorToString(getColor()) + ")"

## 1.8 Class ColorCard extends EffectCard

### 1.8.1 Constructor

Name	Description
+ ColorCard()	เรียก constructor จาก superclass โดย <ul style="list-style-type: none"> <li>- ตั้งค่า number เท่ากับ 14</li> <li>- ตั้งค่า color เป็น Color.BLACK</li> </ul>

### 1.8.2 Method

Name	Description
+ boolean isPlaceable()	ลงได้เสมอ
+ void takeAction()	- ถ้าผู้เล่นคนปัจจุบันเป็นบอท หยุดการเดินเกม 1 วินาที ตั้งค่าความสามารถในการลงการ์ดของผู้เล่นคนต่อไปเป็น false
+ String toString()	คืนค่า "Color Card"

## 1.9 Class ChallengeCard extends EffectCard

### 1.9.1 Constructor

Name	Description
+ ChallengeCard()	เรียก constructor จาก superclass โดย <ul style="list-style-type: none"> <li>- ตั้งค่า number เท่ากับ 15</li> <li>- ตั้งค่า color เป็น Color.BLACK</li> </ul>

### 1.9.2 Method

Name	Description
+ boolean isPlaceable()	ลงได้เสมอ
+ void takeAction()	- ถ้าผู้เล่นคนปัจจุบันเป็นบอท เรียก method challenge() จากผู้เล่นคนปัจจุบัน - ถ้าผู้เล่นคนปัจจุบันเป็นคน ตั้งค่า isColorSelectionState และ isChallengeState ของ GameLogic.getInstance() เป็น true
+ String toString()	คืนค่า "Challenge Card"

## 2. Package entity.player

### 2.1 Abstract Class Player

#### 2.1.1 Field

Name	Description
# String name	ชื่อของผู้เล่น
# ArrayList<UnitCard> cardList	การ์ดที่ผู้เล่นถืออยู่
# ArrayList<UnitCard> placeableCardList	การ์ดที่ลงได้
# boolean isPlayable	ความสามารถในการลงการ์ดของผู้เล่นเมื่อถึงตาตัวเอง
# int score	คะแนนสะสมของผู้เล่น

#### 2.1.2 Constructor

Name	Description
# Player(String name)	ตั้งค่า field name ด้วย name Initialize cardList ตั้งค่า isPlayable เป็น true ตั้งค่า score เป็น 0
# Player()	Initialize cardList

	ตั้งค่า isPlayable เป็น true ตั้งค่า score เป็น 0
--	--

### 2.1.3 Method

Name	Description
+ void <i>play()</i>	เล่นเกมเมื่อถึงตาของตัวเอง
+ void placeCard(UnitCard card)	1. นำ card ออกจาก cardList 2. เพิ่ม GameLogic.getInstance().getCardOnTable() ลงใน GameLogic.getInstance().getCardPile() 3. สับการ์ดใน GameLogic.getInstance().getCardPile() 4. ตั้งค่า cardOnTable ของ GameLogic.getInstance() ด้วย card 5. ถ้า card เป็น EffectCard ให้เรียก method takeAction() จาก card
+ void drawCard(int n)	ย้าย n การ์ดจาก GameLogic.getInstance().getCardPile() มายัง cardList()
+ void <i>challenge()</i>	ทำการชาเลนจ์
+ boolean isWin()	เช็คขนาด cardList() เท่ากับ 0 หรือไม่
+ void setPlaceableCardList()	เพิ่มทุกการ์ดที่ isPlaceable() ใน cardList ลงใน placeableCardList
+ getter & setter for each field	

## 2.2 Class User extends Player

### 2.2.1 Field

Name	Description
- boolean isPlaced	คนเล่นลงการ์ดแล้ว
- boolean isDrawn	คนเล่นจั่วการ์ดแล้ว
- boolean turnEnd	คนเล่นเล่นจบตาแล้ว

### 2.2.2 Constructor

Name	Description
+ User()	เรียก constructor จาก superclass

### 2.2.3 Method

Name	Description
+ void play()	<ol style="list-style-type: none"><li>1. setPlaceableCardList()</li><li>2. รอจนกว่าลงการ์ด1ใบแล้ว หรือ จั่วการ์ด1ใบแล้วแต่ไม่มีการ์ดที่ลงได้ จากนั้นจบตาผู้เล่น</li><li>3. ตั้งค่า isPlaced, isDrawn, turnEnd เป็น false แล้วหยุดการเดินเกม 1 วินาที</li></ol>
+ void placeCard(UnitCard card)	เรียก method จาก superclass ถ้า สีของ card ไม่ใช่ Color.BLACK ให้ตั้งค่า isPlaced เป็น true
+ void drawCard(int n)	เรียก method จาก superclass ถ้าผู้เล่นปัจจุบันเป็นคนเล่น <ul style="list-style-type: none"><li>- setPlaceableCardList()</li><li>- ตั้งค่า isDrawn เป็น true</li></ul>
+ void challenge()	เรียก method ตามลำดับดังนี้ <ul style="list-style-type: none"><li>- GameLogic.getInstance().punishChallenge()</li><li>- GameLogic.getInstance().setChallengeState(false)</li><li>- ตั้งค่า isPlaced เป็น true</li></ul>
+ getter & setter for isPlaced and isDrawn	



## 2.3 Class Bot extends Player

### 2.3.1 Constructor

Name	Description
+ Bot(String name)	เรียก constructor จาก superclass

### 2.3.2 Method

Name	Description
+ void play()	setPlaceableCardList() - ถ้าไม่มีการ์ดที่ลงได้แล้ว ให้จั่วการ์ด 1 ใบ แล้ว setPlaceableCardList() - ถ้ามีการ์ดที่ลงได้ให้ลงการ์ดที่ได้จาก method wisePlace()
+ void placeCard(UnitCard card)	หยุดการเดินเกม 2 วินาที เรียก method AudioLoader.mouseClick1Sound.play() เรียก method จาก superclass
+ void drawCard(int n)	ถ้าผู้เล่นคนปัจจุบันเป็นบอทหรือการ์ดที่อยู่บนโต๊ะไม่ใช่ DrawCard - หยุดการเดินเกม 2 วินาที เรียก method AudioLoader.mouseClick2Sound.play() เรียก method จาก superclass
+ void challenge()	1. หยุดการเดินเกม 2 วินาที 2. เลือกสีที่จะชาเลนจ์จาก method chooseColor() 3. เรียก method setChallengeColor() และ setColorState() จาก GameLogic.getInstance() ด้วยค่าสีที่จะชาเลนจ์ 4. เรียก method ดังนี้ จาก GameLogic.getInstance() - setChallengeState() ด้วยค่า true - punishChallenge() - setChallengeState ด้วยค่า false
- UnitCard wisePlace()	สับการ์ดใน drawableCardList

	<p>สุ่มคืนค่าการ์ด 1 ใบ จาก drawableCardList ตามลำดับ ความสำคัญดังนี้</p> <ol style="list-style-type: none"> <li>1. ถ้าผู้เล่นคนต่อไปเหลือการ์ดในมือ 1 ใบ <ul style="list-style-type: none"> <li>- ChallengeCard</li> <li>- DrawCard</li> <li>- SkipCard หรือ ReverseCard</li> <li>- NormalCard</li> <li>- ColorCard</li> </ul> </li> <li>2. ถ้าผู้เล่นคนต่อไปเหลือการ์ดในมือ 2-4 ใบ <ol style="list-style-type: none"> <li>2.1 ถ้าผู้เล่นคนก่อนหน้ามีการ์ดในมือน้อยกว่าผู้เล่นคนถัดไป <ul style="list-style-type: none"> <li>- ChallengeCard หรือ DrawCard หรือ SkipCard</li> <li>- NormalCard</li> <li>- ReverseCard</li> <li>- ColorCard</li> </ul> </li> <li>2.2 ถ้าไม่ใช่ข้อ 2.1 <ul style="list-style-type: none"> <li>- ChallengeCard หรือ DrawCard หรือ SkipCard หรือ ReverseCard</li> <li>- Normal Card</li> <li>- ColorCard</li> </ul> </li> </ol> </li> <li>3. ถ้าผู้เล่นคนต่อไปเหลือการ์ดในมือ 5 ใบขึ้นไป <ul style="list-style-type: none"> <li>- การ์ดที่ไม่ใช่สีดำ</li> <li>- ColorCard</li> <li>- ChallengeCard</li> </ul> </li> </ol>
+ Color chooseColor()	คืนค่าสีของการ์ดที่บอหมี่ในมือมากที่สุด ถ้าเท่ากันให้สุ่มคืนค่า

### 3. Package gui

#### 3.1 Interface Updatable

Interface ระบุ method ของ Pane ที่มีการเปลี่ยนแปลงตามเกมหรือเวลา

Method

Name	Description
+ void update()	ถูกเรียกทุกๆช่วงเวลาสั้น

#### 3.2 Class FontCardPane extends StackPane

##### 3.2.1 Field

Name	Description
- UnitCard card	การ์ดของ instance

##### 3.2.2 Constructor

Name	Description
+ FontCardPane(UnitCard card)	ตั้งค่า field card ด้วย card ตั้งค่า PrefHeight เป็น 100 ตั้งค่า PrefWidth เป็น 72 เรียก method draw() ตั้งค่า setOnMouseClicked ด้วย method onClickHandler() ตั้งค่า setOnMouseEntered ด้วย method onMouseEnteredHandler() ตั้งค่า setOnMouseExited ด้วย method onMouseExitedHandler()

### 3.2.3 Method

Name	Description
- void onClickHandler()	<p>1. ถ้ารอบการเล่นจบแล้ว หรือ ผู้เล่นปัจจุบันเป็นบอท หรือ card อยู่บนโต๊ะตรงกลาง ให้ return ได้เลย</p> <p>2. ถ้าผู้เล่นที่เป็นคนลงการ์ดไปแล้ว หรือ เกมอยู่ใน color selection state หรือเกมอยู่ใน challenge state ให้สั่ง AudioLoader.nopeSound.play() แล้ว return</p> <p>3. ถ้า card อยู่ใน placeableCardList ของผู้เล่นที่เป็นคน ให้ผู้เล่นนั้นลง card แล้วสั่ง AudioLoader.mouseClick1Sound.play()</p> <p>4. ถ้า card ไม่อยู่ใน placeableCardList ของผู้เล่นที่เป็นคน สั่ง AudioLoader.nopeSound.play()</p> <p>Note : เช็ครอบการเล่นว่าจบแล้วหรือไม่ จาก GameLogic.getInstance().isRoundEnd()</p>
- void onMouseEnteredHandler()	ตั้งค่า cursor เป็น hand เมื่อผู้เล่นที่เป็นคนกำลังเล่น ยังสามารถลงการ์ดได้ และ card ไม่อยู่บนโต๊ะตรงกลาง
- void onMouseExitedHandler()	ตั้งค่า cursor เป็น default
+ void draw()	ใส่สี รูป ตัวเลข ตามประเภท และ field ของ card



Figure 14: FontCardPane

### 3.3 Class BackCardPane extends StackPane

#### 3.3.1 Constructor

Name	Description
+ BackCardPane()	ตั้งค่า PrefHeight เป็น 80 ตั้งค่า PrefWidth เป็น 58 เรียก method draw()

#### 3.3.2 Method

Name	Description
+ void draw()	ใส่สี และ logo ของด้านหลังการ์ด



Figure 15: BackCardPane

### 3.4 Class MenuPane extends VBox

#### 3.4.1 Field

Name	Description
- Button scoreBtn	ปุ่ม SCORE กดเพื่อแสดงตารางคะแนนในขณะนั้น
- Button homeBtn	ปุ่ม HOME กดเพื่อกลับไปสู่ Home Screen
- Button quitBtn	ปุ่ม QUIT กดเพื่อออกจากเกม

#### 3.4.2 Constructor

Name	Description
+ MenuPane()	ตั้งค่า PrefHeight เป็น 200

	<p>ตั้งค่า MaxHeight เป็น 200</p> <p>ตั้งค่า PrefWidth เป็น 150</p> <p>ตั้งค่า MaxWidth เป็น 150</p> <p>ตั้งค่า Spacing เป็น 20</p> <p>ตั้งค่า padding ของขอบล่างเป็น 20</p> <p>ตั้งค่า Alignment ด้วย Pos.BOTTOM_RIGHT</p> <p>Initialize all field</p> <p>setOnAction ของ scoreBtn ด้วย toggleScore()</p> <p>setOnAction ของ homeBtn ด้วย returnToHomeScene()</p> <p>setOnAction ของ quitBtn ด้วย quitGame()</p> <p>setOnMouseEntered และ setOnMouseExited ของทุก field โดยใส่เสียง ปรับตัวอักษร cursor เพื่อความสวยงาม</p>
--	--

### 3.4.3 Method

Name	Description
- void toggleScore()	<p>สั่ง AudioLoader.popSound.play()</p> <p>ถ้ารอบการเล่นยังไม่จบ ให้ตั้งค่า GameLogic.getInstance().isScoreShown() ด้วยค่านิเสธของค่าเดิม</p>
- void returnToHomeScene()	<p>เรียก method ตามลำดับดังนี้</p> <ul style="list-style-type: none"> <li>- AudioLoader.popSound.play()</li> <li>- Main.getRidOfGameRun() เพื่อกำจัด GameRun thread ที่ใช้อยู่ ณ ขณะนั้น</li> <li>- Main.initializeHomeScene() เพื่อสร้าง GameRun thread ใหม่ และสั่ง start()</li> </ul>
- void quitGame()	หยุดการทำงานของโปรแกรม

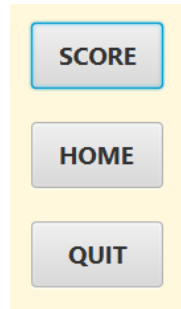


Figure 16: MenuPane

### 3.5 Class UserPane extends VBox implements Updatable

#### 3.5.1 Field

Name	Description
- Player user	ผู้เล่นที่เป็นคน
- GridPane cardPane	Pane ที่แสดงการ์ด ใน cardList ของ user
- Text nameText	ข้อความบอกชื่อ user

#### 3.5.2 Constructor

Name	Description
+ UserPane(Player user)	<p>ตั้งค่า field user ด้วย user</p> <p>ตั้งค่า PrefHeight เป็น 300</p> <p>ตั้งค่า MaxHeight เป็น 300</p> <p>ตั้งค่า PrefWidth เป็น 830</p> <p>ตั้งค่า MaxWidth เป็น 830</p> <p>ตั้งค่า Spacing เป็น 5</p> <p>ตั้งค่า Alignment ด้วย Pos.CENTER</p> <p>Initialize cardPane</p> <ul style="list-style-type: none"> <li>- ตั้งค่า Vgap เป็น 8</li> <li>- ตั้งค่า MaxWidth เป็น 830</li> </ul>



	<ul style="list-style-type: none"> <li>- ตั้งค่า PrefHeight เป็น 260</li> <li>- ตั้งค่า Padding เป็น 10</li> <li>- ตั้งค่า Alignment ด้วย Pos.CENTER</li> <li>- ตั้งค่า Border เป็นสี Tan</li> <li>- ตั้งค่า Background ด้วย ImageLoader.woodTableImg</li> </ul> <p>Initialize nameText</p> <ul style="list-style-type: none"> <li>- ตั้งค่าข้อความเป็น "Player : " + user.getName()</li> <li>- ตั้งค่าขนาดตัวอักษร</li> <li>- setVisible(true)</li> </ul> <p>เพิ่ม nameText และ cardPane ลงใน instance ตามลำดับ เรียก method update()</p>
--	--

### 3.5.3 Method

Name	Description
+ void update()	<p>อัปเดตกรอบของ instance ดังนี้</p> <ol style="list-style-type: none"> <li>1. ผู้เล่นปัจจุบัน ไม่ใช่ user <ul style="list-style-type: none"> <li>- ตั้งค่าสี Border เหมือนตอนเริ่มต้น</li> <li>- ตั้งค่า isVisible ของ nameText เป็น true</li> </ul> </li> <li>2. ผู้เล่นปัจจุบันคือ user และมีความสามารถในการลงการ์ด <ul style="list-style-type: none"> <li>- สลับสี Border ระหว่างสีเริ่มต้นกับสีเขียว</li> <li>- สลับ visibility ของ nameText</li> </ul> </li> <li>3. ผู้เล่นปัจจุบันคือ user แต่ไม่มีความสามารถในการลงการ์ด <ul style="list-style-type: none"> <li>- สลับสี Border ระหว่างสีเริ่มต้นกับสีแดง</li> <li>- สลับ visibility ของ nameText</li> </ul> </li> </ol> <p>เรียก updateCard() เพื่ออัปเดต cardPane</p>
- void updateCard()	ลบทุก children ออก

	สร้าง FontCardPane instance ตามการ์ดทั้งหมดที่ user มี แล้วใส่เข้าไปใน cardPane
--	---



Figure 17: UserPane

### 3.6 Class BotPane extends VBox implements Updatable

#### 3.6.1 Field

Name	Description
- Player bot	ผู้เล่นที่เป็นคน
- GridPane cardPane	Pane ที่แสดงการ์ด ใน cardList ของ bot
- Text nameText	ข้อความบอกชื่อ bot

#### 3.6.2 Constructor

Name	Description
+ BotPane(Player bot)	<p>ตั้งค่า field bot ด้วย bot</p> <p>ตั้งค่า PrefHeight เป็น 150</p> <p>ตั้งค่า MaxHeight เป็น 150</p> <p>ตั้งค่า PrefWidth เป็น 360</p> <p>ตั้งค่า MaxWidth เป็น 360</p> <p>ตั้งค่า Spacing เป็น 5</p> <p>ตั้งค่า Alignment ด้วย Pos.CENTER</p>

	<p>Initialize cardPane</p> <ul style="list-style-type: none"> <li>- ตั้งค่า Hgap เป็น 4</li> <li>- ตั้งค่า MaxWidth เป็น 360</li> <li>- ตั้งค่า PrefHeight เป็น 120</li> <li>- ตั้งค่า Padding เป็น 10</li> <li>- ตั้งค่า Alignment ด้วย Pos.CENTER</li> <li>- ตั้งค่า Border เป็นสี Tan</li> <li>- ตั้งค่า Background ด้วย ImageLoader.woodTableImg</li> </ul> <p>Initialize nameText</p> <ul style="list-style-type: none"> <li>- ตั้งค่าข้อความเป็น "Player : " + bot.getName()</li> <li>- ตั้งค่าขนาดตัวอักษร</li> <li>- setVisible(true)</li> </ul> <p>เพิ่ม nameText และ cardPane ลงใน instance ตามลำดับ เรียก method update()</p>
--	--

### 3.6.3 Method

Name	Description
+ void update()	<p>อัปเดตกรอบของ instance ดังนี้</p> <ol style="list-style-type: none"> <li>1. ผู้เล่นปัจจุบัน ไม่ใช่ bot <ul style="list-style-type: none"> <li>- ตั้งค่าสี Border เหมือนตอนเริ่มต้น</li> <li>- ตั้งค่า isVisible ของ nameText เป็น true</li> </ul> </li> <li>2. ผู้เล่นปัจจุบันคือ bot และมีความสามารถในการลงการ์ด <ul style="list-style-type: none"> <li>- สลับสี Border ระหว่างสีเริ่มต้นกับสีเขียว</li> <li>- สลับ visibility ของ nameText</li> </ul> </li> <li>3. ผู้เล่นปัจจุบันคือ bot แต่ไม่มีความสามารถในการลงการ์ด <ul style="list-style-type: none"> <li>- สลับสี Border ระหว่างสีเริ่มต้นกับสีแดง</li> <li>- สลับ visibility ของ nameText</li> </ul> </li> </ol>

	<p>อัปเดต cardPane</p> <ul style="list-style-type: none"> <li>- ลบทุก children ออก</li> <li>- สร้าง BackCardPane instance ตามจำนวนการ์ดที่ bot มี และไม่เกิน 5 ใบ</li> <li>- ถ้าเกินให้สร้างข้อความบอกจำนวนการ์ดที่ bot ยังมีอีก</li> <li>- ใส่ instance ทั้งหมดที่กล่าวมาใน cardPane</li> </ul>
--	--



Figure 18: BotPane (Of Jesica)

### 3.7 Class BottomPane extends StackPane

#### 3.7.1 Field

Name	Description
- UserPane userPane	กระดานของผู้เล่นที่เป็นคน
- MenuPane menuPane	กระดาน menu
- Button soundBtn	ปุ่มควบคุมเสียง

#### 3.7.2 Constructor

Name	Description
+ BottomPane(UserPane userPane)	<p>ตั้งค่า PrefHeight เป็น 300</p> <p>ตั้งค่า MaxHeight เป็น 300</p> <p>ตั้งค่า PrefWidth เป็น 1200</p> <p>ตั้งค่า MaxWidth เป็น 1200</p> <p>ตั้งค่า field userPane ด้วย userPane</p>

	Initialize menuPane Initialize soundBtn <ul style="list-style-type: none"> <li>- setOnAction ให้เปิดปิดเสียงดนตรีพื้นหลัง และเปลี่ยนรูป icon บนปุ่มให้สอดคล้องกัน</li> <li>- setOnMouseEntered เปลี่ยน cursor ให้เป็น hand</li> </ul> เพิ่ม userPane ลงใน instance ตั้งค่า Alignment เป็น Pos.CENTER เพิ่ม menuPane ลงใน instance ตั้งค่า Alignment เป็น Pos.BOTTOM_RIGHT เพิ่ม soundBtn ลงใน instance ตั้งค่า Alignment เป็น Pos.BOTTOM_LEFT
--	--



Figure 19: BottomPane

### 3.8 Class ColorSelectionPane extends HBox

#### 3.8.1 Field

Name	Description
- Text text	ข้อความบอกให้เลือกสี
- Button redBtn	ปุ่มสีแดง กดเพื่อเลือกสีแดง
- Button yellowBtn	ปุ่มสีเหลือง กดเพื่อเลือกสีเหลือง
- Button greenBtn	ปุ่มสีเขียว กดเพื่อเลือกสีเขียว
- Button blueBtn	ปุ่มสีน้ำเงิน กดเพื่อเลือกสีน้ำเงิน

### 3.8.2 Constructor

Name	Description
+ ColorSelectionPane()	<p>ตั้งค่า MaxHeight เป็น 50</p> <p>ตั้งค่า Spacing เป็น 10</p> <p>ตั้งค่า Alignment ด้วย Pos.CENTER</p> <p>Initialize all fields</p> <p>ตั้งค่าข้อความของ field text ด้วย "Select Color : "</p> <p>ตั้งค่า PrefHeight และ PrefWidth ของทุกปุ่มเป็น 25</p> <p>ตั้งค่า Background ของปุ่มตามสีปุ่ม</p> <p>setOnAction ให้กับทุกปุ่ม โดยเรียก method ดังนี้</p> <ul style="list-style-type: none"> <li>- AudioLoader.buttonClickSound.play()</li> <li>- selectColor() ด้วยสี Background ของปุ่ม</li> </ul> <p>setOnMouseEntered และ setOnMouseExited ให้กับทุกปุ่ม</p> <p>เพื่อเพิ่มความสวยงาม (ใส่เสียง เพิ่มขนาดปุ่ม เปลี่ยนเป็น hand cursor เมื่อ mouse enter รีเซ็ตกลับค่าเริ่มต้น เมื่อ mouse exit)</p>



Figure 20: ColorSelectionPane

### 3.8.3 Method

Name	Description
- void selectColor(Color color)	<p>ถ้าเป็น challenge state</p> <ul style="list-style-type: none"> <li>- ตั้งค่า field challengeColor ของ GameLogic.getInstance() ตาม color</li> <li>- ตั้งค่า field colorState ของ GameLogic.getInstance() ตาม color</li> </ul>

	<ul style="list-style-type: none"> <li>- ตั้งค่า field colorSelectionState ของ GameLogic.getInstance() เป็น false</li> <li>- เรียก method challenge() จาก GameLogic.getInstance().getUser().</li> </ul> <p>ถ้า ไม่เป็น challenge state</p> <ul style="list-style-type: none"> <li>- ตั้งค่า field colorState ของ GameLogic.getInstance() ตาม color</li> <li>- ตั้งค่า field isPlaced ของ user ของ GameLogic.getInstance() เป็น true</li> <li>- ตั้งค่า field colorSelectionState ของ GameLogic.getInstance() เป็น false</li> </ul> <p>Note: challenge state เช็คจาก GameLogic.getInstance().isChallengeState()</p>
--	--

### 3.9 Class TablePane extends HBox implements Updatable

#### 3.9.1 Field

Name	Description
- BackCardPane cardPilePane	Pane ของกองการ์ดที่คว่ำอยู่บนโต๊ะ ไว้สำหรับจั่วการ์ด
- FontCardPane cardOnTablePane	Pane ของการ์ดที่หงายอยู่บนโต๊ะตรงกลาง
- VBox statusPane	Pane ที่บอกทิศทางการวนผู้เล่น และบอกสถานะสีของการ์ดบนโต๊ะ
- boolean isClockwise	ทิศทางการวนผู้เล่นตามเข็มนาฬิกาหรือไม่
- ImageView rotationImg	รูปแสดงทิศทางการวนผู้เล่น



### 3.9.2 Constructor

Name	Description
+ TablePane()	<p>ตั้งค่า MaxHeight เป็น 120</p> <p>ตั้งค่า Spacing เป็น 30</p> <p>ตั้งค่า Alignment ด้วย Pos.CENTER</p> <p>Initialize filed cardPilePane</p> <ul style="list-style-type: none"> <li>- ตั้งค่า PrefWidth เป็น 86</li> <li>- ตั้งค่า PrefHeight เป็น 120</li> <li>- เรียก method draw()</li> <li>- setOnMouseClicked ด้วย method onClickHandler()</li> <li>- setOnMouseEntered ให้เปลี่ยน cursor เป็น hand</li> </ul> <p>เพิ่ม cardPilePane ลงใน instance</p> <p>Initialize field rotationImg โดยเรียก initializerotationImg()</p> <p>อัปเดต instance โดยเรียก update()</p>

### 3.9.3 Method

Name	Description
- void onClickHandler()	<ol style="list-style-type: none"> <li>1. ถ้าวรอบการเล่นจบแล้ว หรือ ผู้เล่นปัจจุบันเป็นบอท return ได้เลย</li> <li>2. ถ้า isPlayable() ของผู้เล่นปัจจุบันที่เป็นคน เท่ากับ false ให้ return ได้เลย</li> <li>3. ถ้าผู้เล่นที่เป็นคนจั่วการ์ดไปแล้ว หรือ ลงการ์ดไปแล้ว หรือ เกมอยู่ใน color selection state หรือเกมอยู่ใน challenge state ให้สั่ง AudioLoader.nopeSound.play() แล้ว return</li> </ol> <p>ถ้าไม่ใช่ข้อ 1-3 เรียกคำสั่ง</p> <ul style="list-style-type: none"> <li>- AudioLoader.mouseClick2Sound.play()</li> <li>- ให้ผู้เล่นที่เป็นคนจั่วการ์ด 1 ใบ</li> </ul>

- void initializeRotationImg()	<p>ตั้งค่า field isClockwise ด้วย method GameLogic.getInstance().isClockwise()</p> <p>ตั้งค่า rotationImg ตาม field isClockwise</p> <p>ตั้งค่า FitHeight และ FitWidth เป็น 50</p>
+ void update()	<p>อัปเดตการเปลี่ยนแปลงของ instance ทุกๆระยะเวลาสั้นๆ ด้วย method</p> <p>initializeCardOnTablePane() และ initializeStatusPane()</p>
- void initializeCardOnTablePane()	<p>ถ้ารอบการเล่นยังไม่จบให้</p> <p>remove cardOnTablePane ออกจาก instance</p> <p>ตั้งค่า field cardOnTablePane ใหม่จาก method getCardOnTable() ของ GameLogic.getInstance()</p> <ul style="list-style-type: none"> <li>- ตั้งค่า PrefWidth เป็น 86</li> <li>- ตั้งค่า PrefHeight เป็น 120</li> <li>- เรียก method draw()</li> </ul> <p>เพิ่ม cardOnTablePane ลงใน instance</p> <p>Note : เช็ครอบการเล่นว่าจบแล้วหรือไม่ จาก GameLogic.getInstance().isRoundEnd()</p>
- void initializeStatusPane()	<p>Remove statusPane ออกจาก instance</p> <p>ถ้ารอบการเล่นยังไม่จบให้</p> <ul style="list-style-type: none"> <li>- สร้าง Rectangle instance ที่มีสีเดียวกับ color state</li> <li>- ถ้า field isClockwise ไม่ตรงกับ ทิศการวนผู้เล่น ให้ เรียก initializeRotationImg()</li> <li>- แต่ถ้าตรงกันให้หมุน rotationImg ตามทิศการวนผู้เล่น ไป 30 องศา</li> <li>- Initialize filed statusPane</li> <li>- ตั้งค่า Spacing ของ statusPane เป็น 25</li> </ul>

	<ul style="list-style-type: none"> <li>- ตั้งค่า Alignment ของ statusPane เป็น Pos.CENTER</li> <li>- เพิ่ม Rectangle instance ที่เพิ่งสร้างขึ้น กับ rotationImg ลงไปใน statusPane ตามลำดับ</li> <li>- เพิ่ม statusPane ลงใน instance</li> </ul> <p>Note:</p> <p>เช็ค color state จาก GameLogic.getInstance().getColorState()</p> <p>เช็ค ทิศการวนผู้เล่น จาก GameLogic.getInstance().isClockwise()</p>
--	--

### 3.10 Class ScorePane extends StackPane

#### 3.10.1 Field

Name	Description
- Text title	Title ของตาราง ถ้ายังไม่มีผู้ชนะ เป็นคำว่า “SCORE BOARD” ถ้ามีผู้ชนะ เป็นข้อความบอกว่าใครชนะ
- GridPane scoreTable	ตารางคะแนน
- Text userPlus	คะแนนที่เพิ่มในรอบนั้นหรือรอบล่าสุดของผู้เล่น(คน)
- Text jessicaPlus	คะแนนที่เพิ่มในรอบนั้นหรือรอบล่าสุดของบอท Jessica
- Text magaretPlus	คะแนนที่เพิ่มในรอบนั้นหรือรอบล่าสุดของบอท Magaret
- Text vandaPlus	คะแนนที่เพิ่มในรอบนั้นหรือรอบล่าสุดของบอท Vanda
- Text userScore	คะแนนรวมของผู้เล่น(คน)
- Text jessicaScore	คะแนนรวมของบอท Jessica
- Text magaretScore	คะแนนรวมของบอท Magaret

- Text vandaScore	คะแนนรวมของบอท Vanda
- Button btn	ปุ่มดำเนินการต่อ ถ้ายังไม่มีใครชนะ จะกดเพื่อเล่นรอบใหม่ ถ้ามีใครชนะ จะกดเพื่อเริ่มเกมใหม่

### 3.10.2 Constructor

Name	Description
+ ScorePane()	<p>ตั้งค่า Padding ด้วย Insets(20, 0, 10, 0)</p> <p>ตั้งค่าสี Background ด้วย Color.STEELBLUE</p> <p>Initialize filed title ด้วยข้อความ "SCORE BOARD"</p> <p>Initialize field scoreTable</p> <ul style="list-style-type: none"> <li>- ตั้งค่า Vgap เป็น 4</li> <li>- ตั้งค่า Hgap เป็น 50</li> <li>- ตั้งค่า Alignment เป็น Pos.CENTER</li> </ul> <p>Initialize ทุก field ที่บอกคะแนนที่เพิ่มของผู้เล่นด้วย “ ”</p> <p>Initialize ทุก field ที่บอกคะแนนรวมของผู้เล่นด้วย “0”</p> <p>สร้าง Text instance ของชื่อผู้เล่นทุกคนและชื่อcolumn</p> <p>เพิ่ม Text instance ที่เพิ่งสร้างและ field ต่างๆลงใน scoreTable</p> <p>เพิ่ม scoreTable ลงใน instance</p> <p>Initialize btn ด้วย Text “Continue”แล้ว setOnAction ดังนี้</p> <ul style="list-style-type: none"> <li>- AudioLoader.mouseClick1Sound.play()</li> <li>- btn.setDisable(true)</li> <li>- ถ้ามีผู้ชนะ</li> </ul> <p>เรียก start() method ของ GameLogic.getInstance() ด้วยชื่อผู้เล่นเดิม</p> <p>สั่ง Main.runGame();</p> <p>สั่ง resetScoreTable();</p>

	<ul style="list-style-type: none"> <li>- ถ้ายังไม่มีผู้ชนะ เริ่มรอบการเล่นใหม่ด้วย method <code>newGame()</code> จาก <code>GameLogic.getInstance()</code></li> <li>- <code>setMouseOnEntered</code> กับ <code>setMouseOnExited</code> คล้ายกับปุ่ม อื่นๆในเกม เพื่อความสวยงาม</li> </ul> <p>เพิ่ม btn ใน instance ตั้งค่า disable ของ btn เป็น true</p> <p>Note: เช็ค ผู้ชนะ จาก <code>GameLogic.getInstance().getWinner()</code></p>
--	---

### 3.10.3 Method

Name	Description
+ void <code>setScore()</code>	<p>นับ score จากจำนวนการ์ดที่เหลือใน กระดานของผู้เล่น ทั้งหมด</p> <p>เพิ่ม score ให้กับผู้เล่นที่ชนะ</p> <p>ตั้งค่า Text ใหม่ สำหรับfield ที่จำเป็น เพื่ออัปเดต scoreTable</p> <p>ถ้ามีผู้ชนะ</p> <ul style="list-style-type: none"> <li>- เปลี่ยนค่า Text ของ title เป็นชื่อผู้ชนะ + “ WIN !”</li> <li>- เปลี่ยนชื่อ btn เป็น “New Game”</li> </ul> <p>ตั้งค่า disable ของ btn เป็น false</p>
- void <code>resetScoreTable()</code>	<p>ตั้งค่า Text ของทุก field ยกเว้น scoreTable ให้เหมือนตอน เรียก Constructor</p>



Figure 21: score board

### 3.11 Class CenterPane extends StackPane implements Updatable

#### 3.11.1 Field

Name	Description
- TablePane tablePane	โต๊ะตรงกลาง แสดงการ์ดที่หงาย กองการ์ด สถานะสี ทิศการวนผู้เล่น
- ColorSelectionPane colorSelectionPane	Pane สำหรับคนได้เลือกสี เมื่อลงการ์ดที่ต้องเลือกสี วางอยู่ส่วนล่างของ tablePane
- Text challengeResult	ผล challenge เมื่อมีการลงการ์ดชาเลนจ์ วางที่ตำแหน่งเดียวกันกับ colorSelectionPane
- ScorePane scoreBoard	ตารางบอกคะแนนปัจจุบันของทุกผู้เล่น

#### 3.11.2 Constructor

Name	Description
+ CenterPane()	ตั้งค่า PrefHeight เป็น 260 ตั้งค่า MaxHeight เป็น 260 ตั้งค่า Padding ด้วย Inset(10, 0, 10, 0) ตั้งค่า Alignment ด้วย Pos.CENTER ตั้งค่าสี Background เป็น Color.LIGHTSTEELBLUE

	Initialize field tablePane แล้วเพิ่มลงใน instance (Pos.CENTER) Initialize field colorSelectionPane แล้วเพิ่มลงใน instance (Pos.BOTTOM_CENTER) Initialize field challengeResult แล้วเพิ่มลงใน instance (Pos.BOTTOM_CENTER) Initialize field scoreBoard แล้วเพิ่มลงใน instance (Pos.CENTER) Set scoreBoard to invisible เรียก method update()
--	--

### 3.11.3 Method

Name	Description
+ void update()	<p>อันดับแรก อัปเดต scoreBoard</p> <ul style="list-style-type: none"> <li>- ถ้ารอบการเล่นจบแล้วแล้วยังไม่ได้คิดคะแนนใหม่ ให้คิดคะแนนใหม่ จากนั้นแสดงตารางคะแนนขึ้นมา</li> <li>- ตั้งค่า visible ของ scoreBoard ตาม method isScoreShown() ของ GameLogic.getInstance()</li> </ul> <p>ต่อมาเป็นการอัปเดต tablePane</p> <ul style="list-style-type: none"> <li>- เรียก method tablePane.update()</li> <li>- ถ้ารอบการเล่นจบแล้ว ตั้งค่า visible เป็น false ไม่เช่นนั้น ตั้งค่าเป็น true</li> </ul> <p>สุดท้ายตั้งค่า colorSelectionPane และ challengeResult ด้วย setColorSelectionAndChallengeResult()</p>
- void setColorSelectionAndChallengeResult()	<p>แสดง colorSelectionPane เมื่อคนเล่นต้องเลือกทำงานเลือกสี</p> <p>ถ้าเป็นการเลือกสีเพื่อชาเลนจ์ให้แสดงผลชาเลนจ์หลังจากเลือกสีเสร็จแล้ว</p>



	ถ้าไม่อยู่ในสถานะเลือกสีหรือชาเลนจ์ ให้ set visible ของทั้ง colorSelectionPane และ challengeResult เป็น false
--	---

## 4. Package logic

### 4.1 Class GameLogic

เป็นศูนย์กลางการเก็บสถานะของเกม ผู้เล่นทั้งหมด เป็นแหล่งอ้างอิงในการดำเนินเกม การเปลี่ยนแปลงของหน้าจอ และการเล่นโดยอัตโนมัติของบอท

#### 4.1.1 Field

Name	Description
- <u>GameLogic</u> instance	Instance เพียงหนึ่งเดียวของ Class GameLogic เพื่อให้มั่นใจว่าเกมจะดำเนินโดยไม่มีข้อมูลเกมและผู้เล่นทับซ้อนกัน
- Color[] colorArray	Array ของสีการ์ดหลัก 4 สี ได้แก่ น้ำเงิน เขียว แดง เหลือง
- ArrayList<UnitCard> cardPile	ArrayList ของการ์ดที่เอาไว้จั่ว
- UnitCard cardOnTable	การ์ดบนโต๊ะตรงกลางที่ผู้เล่นจะต้องลงการ์ดให้สอดคล้องกับสีหรือตัวเลข
- int playerTurn	ตัวเลขที่ระบุตาผู้เล่น
- Player nextPlayer	ผู้เล่นปัจจุบัน(ที่กำลังเล่นอยู่)
- Player beforePlayer	ผู้เล่นคนก่อนหน้า
- int numberState	ผู้เล่นคนถัดไป
- Color colorState	สถานะสีของการ์ดที่อยู่บนโต๊ะ(ตรงกลาง)
- boolean isClockwise	ทิศการวนผู้เล่นว่าตามเข็มนาฬิกาหรือไม่
- boolean isRoundEnd	ระบุว่าการรอบการเล่นรอบนี้จบแล้วหรือยัง
- boolean colorSelectionState	ระบุว่ายู่ในสถานะที่คนเล่นกำลังเลือกสีหรือไม่

- boolean challengeState	ระบุว่าอยู่ในสถานะที่ผู้เล่น(ทั้งคนและบอท)กำลังหาเลนจ์ผู้เล่นคนถัดไปหรือไม่
- Color challengeColor	ระบุสีที่ผู้เล่นคนปัจจุบันเลือกใช้หาเลนจ์
- boolean isScoreShown	ระบุว่าโชว์กระดานคะแนนหรือไม่
- Player user	ผู้เล่นที่เป็นคนเล่น
- Player botJesica	ผู้เล่นบอท ชื่อ Jesica
- Player botMagaret	ผู้เล่นบอท ชื่อ Magaret
- Player botVanda	ผู้เล่นบอท ชื่อ Vanda
- int deckSize	จำนวนการ์ดทั้งหมดที่มีในเกม

#### 4.1.2 Constructor

Name	Description
- GameLogic()	Initialize user, botJesica, botMagaret, botVanda

#### 4.1.3 Method

Name	Description
<u>+ GameLogic getInstance()</u>	ถ้า instance == null - Initialize instance Return instance
+ void start(String name)	Method ที่ถูกเรียกเมื่อเริ่มเกมใหม่ ตั้งชื่อ user ด้วย name ตั้งค่าคะแนนของทุกผู้เล่นให้เป็น 0 เรียก newGame() เพื่อเริ่มรอบการเล่นใหม่
+ void newGame()	ลบการ์ดที่มีทั้งหมดในเกมออก เรียก initilizeCardPile() เรียก dealCard() ตั้งค่า field ต่างๆเพื่อเริ่มเกมรอบใหม่ ดังนี้ - setNumberState(cardOnTable.getNumber()) - setColorState(cardOnTable.getColor())

	<ul style="list-style-type: none"> <li>- setClockwise(true)</li> <li>- setRoundEnd(false)</li> <li>- setColorSelectionState(false)</li> <li>- setChallengeState(false)</li> <li>- setScoreShown(false)</li> <li>- ตั้งค่า <b>playerTurn</b> โดยการสุ่มตัวเลข</li> <li>- setCurrentPlayer()</li> <li>- setNextPlayer()</li> <li>- setBeforePlayer()</li> <li>- ตั้งค่า <b>isPlayable()</b> ของทุกผู้เล่นให้เป็น true</li> </ul>
- void initializeCardPile()	สร้างการ์ดใหม่ทั้งหมดแล้วใส่ลงใน <b>cardPile</b> พร้อมตั้งค่า <b>deckSize</b>
- void dealCard()	แจกการ์ดแบบสุ่มจาก <b>cardPile</b> ไปให้ผู้เล่นแต่ละคน คนละ 5 ใบ
+ void setUpForNewTurn()	ตั้งค่า <b>numberState</b> จาก <b>number field</b> ของ <b>cardOnTable</b> ถ้า <b>cardOnTable</b> ไม่ใช่การ์ดที่เปลี่ยนสีได้ ให้ตั้งค่า <b>colorState</b> จาก <b>color field</b> ของ <b>cardOnTable</b> <b>move()</b> <b>setCurrentPlayer()</b> <b>setNextPlayer()</b> <b>setBeforePlayer()</b>
- void move()	ถ้าวินผู้เล่นแบบตามเข็มนาฬิกา บวก <b>playerTurn</b> ด้วย 1 ถ้าไม่ใช่ ลบ <b>playerTurn</b> ด้วย 1
+ boolean isChallengeWin()	ถ้าผู้เล่นคนต่อไปมีการ์ดที่มีสีเดียวกับ <b>challengeColor</b> return <b>false</b> นอกจากนั้น return <b>true</b>
+ void punishChallenge()	เรียก <b>method isChallengeWin()</b> ถ้าค่าเป็น true ผู้เล่นคนต่อไปจั่วการ์ด 4 ใบ ถ้าค่าเป็น false ผู้เล่นคนปัจจุบันจั่วการ์ด 2 ใบ
+ Player getWinner()	ถ้ามีผู้เล่นที่มีคะแนนมากกว่าหรือเท่ากับ 24 ให้คืนค่าผู้เล่นคนนั้น ถ้าไม่มีให้คืน <b>null</b>
+ void sleepOne()	เรียก <b>TimeUnit.SECONDS.sleep(1)</b>

+ void sleepTwo()	เรียก TimeUnit.SECONDS.sleep(2)
+ void sleepThree()	เรียก TimeUnit.SECONDS.sleep(3)
+ getter & setter for each field	

## 4.2 Class UpdatableHolder

เป็นที่เก็บ object ที่ต้องอัปเดตตลอดเวลาสั้นๆ

### 4.2.1 Field

Name	Description
- <u>UpdatableHolder</u> <u>instance</u>	Instance เพียงหนึ่งเดียวของ Class UpdatableHolder เพื่อให้มั่นใจว่าเกมจะอัปเดตจากแหล่งข้อมูลเดียว
- ArrayList<Updatable> entities	Updatable object ที่ต้องอัปเดตทั้งหมด

### 4.2.2 Constructor

Name	Description
- UpdatableHolder()	Initialize entities

### 4.2.3 Method

Name	Description
+ static <u>Updatable</u> <u>getInstance()</u>	ถ้า instance == null - Initialize instance Return instance
+ void updateScreen()	เรียก method update() จากทุก object ที่อยู่ใน entities
+ setter for field entities	

#### 4.3 Class GameRun extends Thread

สร้างขึ้นเพื่อเล่นเกมโดยอัตโนมัติ thread หยุดทำงานเมื่อจบเกม (มีผู้ชนะ ได้แต้มครบ 24 แต้ม)

### 5. Package main

#### 5.1 Class Main extends Application

##### 5.1.1 Field

Name	Description
- <u>Stage primaryStage</u>	Stage ของเกม
- <u>Parent homeScene</u>	องค์ประกอบทั้งหมดของ Home screen
- <u>GameScene gameScene</u>	องค์ประกอบทั้งหมดของ Game screen
- <u>GameRun gameRun</u>	Thread ที่ใช้รันในแต่ละเกม

##### 5.1.2 Constructor

Name	Description
+ <u>void main(String[] args)</u>	launch(args)

##### 5.1.3 Method

Name	Description
+ void start(Stage primaryStage) throws Exception	สร้าง Timeline เพื่ออัปเดตหน้าจอทุก 0.4 วินาที ตั้ง Cycle Count เป็น Indefinite เรียก initializeHomeScene()
+ <u>void initializeHomeScene()</u>	สร้าง homeScene จาก HomeScene.fxml Set scene ของ primaryStage จาก scene ที่สร้างด้วย homeScene

	Set title ของ primaryStage เป็น "UNO 24   presented by chomchaby" Set resizable to false สั่ง primaryStage.show() จากนั้นตั้งค่าเสียงพื้นหลังด้วย AudioLoader.gamePlayBGSound.stop(); AudioLoader.startBGSound.play();
<u>+ void</u> <u>initializeGamePlayScene()</u>	Initialize field gameScene Set scene ของ primaryStage จาก scene ที่สร้างด้วย gameScene สั่ง primaryStage.show() สั่ง runGame() จากนั้นตั้งค่าเสียงพื้นหลังด้วย AudioLoader.startBGSound.stop(); AudioLoader.gamePlayBGSound.play();
<u>+ void</u> <u>runGame()</u>	Initialize gameRun สั่ง gameRun.start()
<u>+ void</u> <u>getRidOfGameRun()</u>	Interrupt gameRun Stop gameRun

## 6. Package screen

### 6.1 Class GameScene extends BorderPane

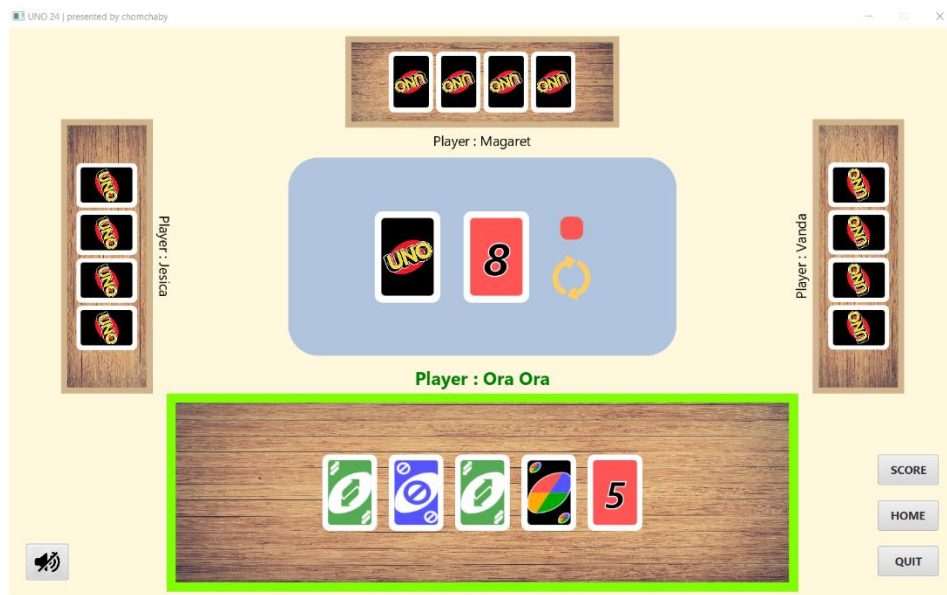


Figure 22: GameScene

#### 6.1.1 Constructor

Name	Description
+ GameScene()	<p>ตั้งค่า PrefWidth เป็น 1250</p> <p>ตั้งค่า PrefHeight เป็น 750</p> <p>ตั้งค่า Padding ด้วย new Insets(10)</p> <p>ตั้งค่าสี Background เป็น Color.CORNSILK</p> <p>สร้าง BotPane ให้กับบอททั้ง 3 ตัว ใส่ลง instance ที่ด้านซ้าย ด้านบนและด้านขวาของ BorderPane ตามรูปด้านบน</p> <p>สร้าง UserPane ให้กับคนเล่น ใส่ลงด้านล่างของ BorderPane</p> <p>สร้าง CenterPane ใส่ลงตรงกลางของ BorderPane</p> <p>สร้าง ArrayList&lt;Updatable&gt; ที่บรรจุ Pane ทั้งหมดที่เพิ่งสร้าง จากนั้นส่ง UpdatableHolder.getInstance().setEntities() โดยส่งค่า ArrayList &lt;Updatable&gt; instance ที่เพิ่งสร้าง</p>

### 6.1.2 Method

Name	Description
+ void start(Stage primaryStage) throws Exception	สร้าง Timeline เพื่ออัปเดตหน้าจอทุก 0.4 วินาที ตั้ง Cycle Count เป็น Indefinite เรียก initializeHomeScene()
+ void <u>initializeHomeScene()</u>	สร้าง homeScene จาก HomeScene.fxml Set scene ของ primaryStage จาก scene ที่สร้างด้วย homeScene Set title ของ primaryStage เป็น "UNO 24   presented by chomchaby" Set resizable to false สั่ง primaryStage.show() จากนั้นตั้งค่าเสียงพื้นหลังด้วย AudioLoader.gamePlayBGSound.stop(); AudioLoader.startBGSound.play();
+ void <u>initializeGamePlayScene()</u>	Initialize field gameScene Set scene ของ primaryStage จาก scene ที่สร้างด้วย gameScene สั่ง primaryStage.show() สั่ง runGame() จากนั้นตั้งค่าเสียงพื้นหลังด้วย AudioLoader.startBGSound.stop(); AudioLoader.gamePlayBGSound.play();
+ void <u>runGame()</u>	Initialize gameRun สั่ง gameRun.start()
+ void <u>getRidOfGameRun()</u>	Interrupt gameRun Stop gameRun



## 6.2 File HomeScene.fxml

FXML file สำหรับ Home screen

## 6.3 Class HomeSceneController extends implements Initializable

Controller สำหรับ HomeScene.fxml

## 7. Package sharedObject

Package สำหรับเก็บรูปภาพ เสียง และสี ที่ใช้ใน project